# The Future of Software and Software Research: Panel Summary

# Where Are We Now?

- Tremendous gains in productivity
- Successful adaptation to new computational models
  - batch vs. interactive
  - distributed
- Ability to easily develop 100,000+ LOC systems
- SW accepted part of safety-critical applications
- Repeatable best practices for many classes of systems
  - automated solutions for limited domains

# Why Can't We Declare Victory?

- Pervasiveness of SW
  - critical infrastructure relies on SW
- Rapid pace of change
  - computational models, technology, marketplace
- Increasing expectations
  - systems of systems, massive concurrency, global connectedness, self-adapting systems, unpredictable integration requirements
- New componentry is poorly-understood
  - COTS, autonomous agents
- Slow adoption of best practices

# Challenges

- Adaptability
- Distribution and concurrency
- Expect and cope with failure
- Platform heterogeneity
- Change propagation
- Component portability
- Program scale
- Richer computing environments
- Integration

# Software Development in Tomorrowland

- Wide-ranging computing environment
- Ever-persistent legacy components
- Rapid development and experimentation
- Wide-ranging modalities
  - speech, real-world sensors, location-aware
- Multi-disciplinary, geographically-distributed developers
  - Stronger domain and business models, involves biologist, psychologists, end-users, team-building aids
- Analysis - automated support
  - agents, design critics, software models, semantics-aware tools
- Persistent, common, refreshable knowledge

# Research Strategies

- Rework / development SE techniques that are
  - value-conscious, people-conscious, change aware, non-uniformly applied
- Conduct large-scale empirical evaluation
  - systems that involve people - modeling human interaction
  - software archeology
- Incorporate meta-qualities into development languages
  - trust management, security, power mgmt., timeliness
- Study actual development processes
  - collaborative s/w development
- Leverage increase in machine power

# Research Strategies (cont.)

- Bridge gap between design and code
  - Make design/designer a first-class object (CAD/CAM)
  - Domain-specific models
- Explore new paradigms
  - biological, economic
- User-accessible techniques
  - Multi-stakeholder specification techniques
  - Safe end-user programming
- Improved SE education and training techniques
  - accelerated learning - immersive techniques
  - multiple levels of IT personnel
- Science of components and integration

# Why we need $5B per year

- Ambitious, broad research agenda
- Need to scale up our research to match new scale of SW
- Need large-scale, multi-domain testbeds
  - SW archeology
    - need many artifacts: reqs., designs, code, test cases, test case outputs, CMS logs
  - Sponsor open-source software projects as a research enabler
  - Encourage sponsored projects to instrument
- Industry pilot projects for promising research technology
- interdisciplinary projects
- Collaborating with other gov't funded research
  - physics, bio, other gov't contractors, etc.
- Need larger research staff
- Must produce more professionals

# The Future (all this for **only** $5B)

- Quality S/W solutions for complex domains such as:
  - battlefield control
  - power grid mgmt
  - automobile traffic management
  - interconnected medical devices
  - emergency services
  - crisis management
  - etc.
- with the following properties
  - affordable
  - reliable, safe
  - easy to change/ very long-lived
  - useable
  - improves quality of life (QoL)
  - etc.